COAL LAB MANUAL

**Submitted to:**

Dr. Tauqir

Teacher assistant : Sir Shahzad

**Submitted By:**

Hamza Farooq

Registeration no : 2016-CS-122

**Section:**

Section B

**Department:**

Department of Computer Science

**Institute:**

University of Engineering and Technology Lahore (Main Campus)



**Date of Submission:** March 29, 2018

**COAL Lab 4**

**IA-32 Assembly Language Sections**

**Description**

In this lab we will learn about

* Language Sections.
* Assembly Directives
* Handling Strings in Assembly Language
* If / else alternatives in Assembly Language.
* Labels and their importance
* Conditional and Unconditional jumps

**Syntax**

**.MODEL:** An Assembly Language directive used to define a memory

model that may be **TINY, SMALL, COMPACT** and so on.

**.DATA:** The data section is used for declaring initialized data or

constants.

**.CODE:** Code section contain all decisions that program actually make

**.EXIT:** An Assembly Language directive used to invoke ExitProcess

function.

**.LABEL:** A symbolic label consists of an identifier followed by a colon

**.CMP:** Compares the first source operand with the second source

operand.

**JNE:** Jump not equal (a conditional jump).

**JE:** Jump equal (a conditional jump).

**JMP:** An unconditional jump.

**LEA:** (load effective address) is often used for loading string.

**Lab Work**  
Write an assembly language program that prompts with a suitable string message to enter a numeric character and display a message that is about if the number is even or odd. Program should ask for run again or not. User have to enter ‘Y’ or ‘N’ keys for run again or exit program respectively.

**Program**

.MODEL SMALL

.data

STR1 DB 10, 13, 'Enter Number: $'

STR2 DB 10, 13, 'Value is Even$'

STR3 DB 10, 13, 'Value is Odd$'

STR4 DB 10, 13, 'Do you want to run again (Y/N)?$'

.code

mov ax, @data

mov DS, AX

START:

LEA DX, STR1

mov ah, 9

int 21h

mov ah,1

INT 21h

mov bl, 2

div bl

mov dl, ah

CMP dl, 0

JNE ODD

EVEN:

LEA DX, STR2

mov ah, 9

int 21h

JMP LAST

ODD:

LEA DX, STR3

mov ah, 9

int 21h

JMP LAST

LAST:

LEA DX, STR4

mov ah, 9

int 21h

mov ah,1

INT 21h

CMP al, 079h

JE START

CMP al, 079h

JE EXIT

EXIT:

.exit

ret